GENDEX

BIB

ALPHA

CIB

CAR

RRC

GENDEX FOR CONSTRUCTING EXPERIMENT DESIGNS

by

W. T. Federer and S. Nshinyabakobeje (Biometrics Department, Cornell University),

and

N-K Nguyen (Mathematical & Information Sciences, CSIRO, Australia),

ABSTRACT

A computer toolkit (package), Gendex, for constructing randomized plans of various types of experiment designs has been developed. For incomplete block experiment designs (IBDs), there are three modules named BIB, ALPHA, and CIB for constructing the plans. For the BIB module, the designs are constructed by using an iterative procedure, which produces an optimal, or near optimal plan. For ALPHA, an alpha array is constructed first and from this the IBD is obtained. A cyclical construction method is employed for the plans obtained from CIB. The iterative process used for these designs provides optimal or near optimal plans. The plans obtained from the different modules may differ in efficiency and optimality properties. For row-column designs, e.g. Latin square and Youden design types, the module CAR is used. An IBD is obtained as a first step and then the row-column design results. To obtain resolvable (treatments arranged in complete blocks) row-column plans, the module RRC is employed. A resolvable IBD is obtained as a first step and then the RRC plan is obtained. These row-column plans are obtained by an iterative procedure resulting in variance optimal or near optimal plans. The installation procedure is described first. A set of commands is given for two different examples for each of the five modules. The printed outputs of randomized plans are presented. Some remarks on various aspects of the Gendex toolkit are presented.

BU-1433-M September 1998

In the Technical Report Series of the Biometrics Department, Cornell University, Ithaca, New York 14853

GENDEX FOR CONSTRUCTING EXPERIMENT DESIGNS

1. INTRODUCTION

Gendex is a software toolkit, which will produce a randomized plan for the following experiment designs for v treatments in incomplete blocks of size k in b incomplete blocks and r replicates using one of the following subroutines:

- Balanced incomplete block design (BIB): $v = kb \leq 100$, $k \leq \min(20, v)$, v divisible by k for resolvable designs (treatments arranged in complete blocks) and rv divisible by k for non-resolvable designs.

- Alpha design (ALPHA): $v \leq 500$, $k \leq 20$, $r \leq 10$, $k \leq v$, $s \leq 50$ for $v = ks$.

- Cyclic design (CIB): $v \leq 500$, $k \leq \min(20, v)$, $\min(v, b) \leq 256$), rv divisible by k.

Also in the toolkit is a module for constructing row-column designs. To obtain optimal or near optimal row-column experiment designs, one of the following modules may be used

- Row-column (k rows, b columns) design (CAR): $v \leq 100$, $k \leq 25$, $b \leq 50$.

- Resolvable row-column design (RRC): $v = ks \leq 100$, $k \leq 20$, $rk \leq 100$.

The best or nearly best incomplete block design (IBD) or row-column design (RCD) for r replicates is obtained by an iterative procedure.

The toolkit also includes modules for fractional replicates of factorials and response surface treatment designs. Fractional replicates of complete factorial designs and response surface designs may be obtained from the modules:

- D-optimal fractional replicate design (DEX): $m \leq 11$ factors for two-level factors and $m \leq 8$ for three-level factors, n = number of runs, $n > p - 1$ for $p = m + 1$ for 2-level Resolution III designs, $p = 1 + m(m + 1)/2$ for 2-level Resolution V designs, and $p = 1 + m + m(m + 1)/2$ for second order response surface designs (RSD).

- Blocking fractional factorials and response surface designs (CUT): n runs in b blocks for $p - b$ variables (m input variables and $p - b - m$ derived variables).

- Trend-free fractional factorials and response surface designs (RAT): n runs with m_0 columns representing trends ($m - m_0$ input variables and $p - 1 - m$ derived variables).

- Fractional factorials and response surface designs (FFD): m variables in n runs, $m < 16$ and $n < 129$.

Included in the package is a program for constructing orthogonal or near-orthogonal arrays. This module is:

- Orthogonal or near-orthogonal arrays (NOA): $m \leq 64$ and $n \leq 64$.

It is possible to use NOA to construct near-orthogonal arrays for factors with different numbers of levels (Nguyen, 1996).

For the modules BIB, ALPHA, CIB, CAR, and RRC, the appropriate commands and two small examples are presented in the following sections. But first we discuss the installation process for the toolkit.

2. INSTALLATION

The toolkit is a DOS package and needs to be installed in the DOS area on a computer. For Windows 95 and using a disk, proceed as follows:

Select MS-DOS Prompt from START menu. Then type

C:\ md Gendex (This names the directory.)
C:\Gendex >A:
A:\ COPY *.* C: (This puts Gendex in DOS on the C drive in the directory named Gendex.)
A:\ C:
C:\Gendex > (This checks that Gendex has been copied on the C drive.)

For Windows 3.1 or higher, proceed as follows:

Windows 3.1 opens in the C drive of DOS. Then type

C>CD\WINDOWS
C>md Gendex
C>Gendex >A:
A> COPY *.* C:
A>C:
C:\Gendex > (This may be C> on some 3.1 installations.)

Type EXIT to leave Gendex.

If the toolkit is obtained as an attachment to an e-mail, the program may be installed by finding where the attachment is stored, bawin\winba\eudora\attach\gendex.exe in our case as Cornell University uses Bear Access and Eudora for e-mail. The location is listed when one clicks on the attachment in the e-mail. Using Windows Explorer, open (double click on) bawin, winba, eudora, and attach in this order. All e-mail attachments are listed. Highlight the attachments desired (use shift key and mouse), alpha to noa.tex for the Gendex attachment. From File select Save As and 3.5 disk. This saves the attachment to disk, and then proceed as above to load program in MS-DOS.

3. BALANCED INCOMPLETE BLOCK DESIGN (BIBD) MODULE

To illustrate the procedure for this module, use v = 9, k = 3, r = 4, random seed = xxx, tries = xx. Use the following commands to start the module from DOS (No spaces are allowed before or after the = sign. The random seed xxx need not be included as the program automatically selects a seed. The number of tries xx may also be omitted and the program automatically sets the number. However, it is suggested that the number of tries be used in all cases to conserve the amount of time required to obtain a plan. The program may be stopped at any time by pressing any key.):

C:\ CD GENDEX
C:\ Gendex> BIB v=9 k=3 r=4 –r SEED=123 TRIES=10

The -r command sorts the BIBD into complete blocks, i.e., a resolvable experiment design. Since a BIBD is obtained with r = 4 replicates, the design is obtained with few tries. We could have used 5 or less as the number of tries to obtain this IBD. To display the output on the monitor, the command is

C:\ Gendex> TYPE BIB.OUT

The incomplete blocks are the rows of the design. If it is desired to have the incomplete blocks in columns, use the command -c. The -r and -c may be put anywhere in the command statement. To print out the randomized form of the plan, use the command:

C:\ Gendex>COPY BIB.OUT PRN (or PRINT BIB.OUT)

Note: For some computer set-ups, it may be necessary to use COPY BIB.OUT PRN and copy the output into some program such as MICROSOFT WORD and print the output from there. This command sends the design to the printer. The resulting printed output for (v,k,r) = (9,3,4) is:

```
NOTE: Construct an IBD with v=9, k=3, r=4 and b=12

rand seed            E        E/U    l l    n1    n2    n3    n4
1874560893      750000      100.00 1       36

NOTE: Best design obtained at try 1.
        5    3    1
        2    7    4
        8    9    6

        1    4    6
        3    8    2
        9    7    5

        5    6    2
        3    9    4
        7    8    1

        2    1    9
        7    6    3
        8    5    4

NOTE: Blocks are rows.
NOTE: BIB used 0.17 seconds.
```

The values of v, k, r, and b used are presented. The random seed, 1874560893, used for the randomization is printed. 10^6 (1E+6) times the design intrablock efficiency factor E = (r – 1)/r = 0.75, i. e. 750000, is printed next (An effect is confounded with incomplete blocks in one of the r replicates and unconfounded in the other r – 1 replicates.). The ratio 100E/U is given, where U is the upper bound for the incomplete block design (IBD). The value under ll is the smallest concurrence value for the design. Here it is one as every treatment occurs once with every other treatment in an incomplete block of the design, i. e., this is a balanced incomplete block design. As noted, the optimal design, E/U = 100.00, was obtained on the first try. The program stops when E/U = 100.00. The value n1 = 36 is the number of concurrences for the lowest concurrence number. Since this is the only concurrence number, there are no values for n2, n3, or n4. The sum n1 + n2 + n3+ n4 = v(v – 1)/2 = 36.

For a second example, consider v = 12, k = 3, r = 4, and tries = 50. The output obtained from the command COPY BIB.OUT PRN was:

```
NOTE: Construct an IBD with v=12, k=3, r=4 and b=16

rand seed          E           E/U    ll    n1    n2    n3    n4
855216484     709677          100.00 0      18    48

NOTE: Best design obtained at try 45.

        8    11    4
        7    2     6
        10   12    3
```

4	7	9
8	3	6
10	5	2
12	11	1
11	5	6
1	7	3
9	8	10
12	4	2
12	6	9
10	7	11
2	1	8
5	4	3

NOTE: Blocks are rows.
NOTE: BIB used 1.32 seconds.

Using only 20 tries resulted in E\U = 99.39. Since E/U was not equal to 100.00, more tries were needed. For this design, ll = 0 for 18 pairs. A concurrence value of one was obtained for the remaining 48 pairs of treatments. No pair of treatments occurred together in an incomplete block more than once. 18 + 48 = 66 = v(v – 1)/2.

4. ALPHA DESIGN MODULE

To illustrate this module, construct an α-design (IBD) for v = 9, k = 3, and r = 4. From Gendex, use the command

ALPHA v=9 k=3 r=4

The number of tries could have been included as well as the random seed. Then use this command to display the design on the monitor

TYPE ALPHA.OUT

To obtain a printed copy of the design, use the command

COPY ALPHA.OUT PRN (or PRINT ALPHA.OUT)

The printed output is

```
NOTE: Alpha design with v=9, k=3, r=4 and b=12

 rand_seed         E     E/U  l1    n1    n2    n3    n4
 298449135    705882  94.12    0     9    18     9

NOTE: Alpha array (r x k) obtained at try 1:

  2  2  1
  2  0  0
  1  0  1
  0  0  1

NOTE: Alpha array was used to construct this design:
```

```
  9   4   1
  2   7   5
  8   3   6

  7   3   4
  8   5   1
  9   6   2

  8   4   2
  7   6   1
  5   9   3

  1   8   4
  3   7   6
  9   2   5

NOTE: Blocks are rows.
NOTE: ALPHA used 2.48 seconds.
```

The lowest concurrence value was for λ_1 equal zero and there were n1 = 9 pairs with this value. There were 18 pairs with the concurrence value of λ_2 equal one, i.e., n2 = 18, and 9 pairs with the concurrence value of λ_3 equal two, i.e., n3 = 9. This design is a three associate class design and its efficiency factor E = 0.705882 is not as at the maximum value of E = 0.75 as for the BIBD. The particular design obtained from BIB is preferred to the one obtained from ALPHA.

Using the same second BIB example as above, start with the commands

ALPHA v=12 k=3 r=4 –r tries=100
COPY ALPHA.OUT PRN (or PRINT ALPHA.OUT)

to obtain the following output:

```
NOTE: Alpha design with v=12, k=3, r=4 and b=16

 rand_seed        E     E/U  l1   n1   n2   n3   n4
1723201820   691824  97.48    0   22   40    4

NOTE: Alpha array (r x k) obtained at try 1:

  3  3  1
  1  0  2
  1  2  1
  0  2  3

NOTE: Alpha array was used to construct this design:

   4  10   8
   2  12   6
   7   9   3
   5  11   1

   4   9   7
  11   5   2
   8   1  10
   3   6  12
```

```
12    4    5
 2   10    7
 6    9    1
11    8    3

11    4    6
 1    7   12
10    5    3
 8    9    2

NOTE: Blocks are rows.
NOTE: ALPHA used 3.90 seconds.
```

A three concurrence design was obtained with $\lambda_1 = 0$, $\lambda_2 = 1$, and $\lambda_3 = 2$. For the BIB design, there were only two concurrences, i.e., zero and one, and hence a more efficient design for which E = 0.709677 versus the above value of E = 0.691824 for this design. n1 + n2 + n3 + n4 = 22 + 40 + 4 = 66 = 12(12 – 1)/2 = v(v – 1)/2.

5. CYCLICAL DESIGN MODULE

The CIB module produces cyclical incomplete block designs, which are not resolvable designs, i.e., the -r command does not operate in this module. When r = ck, c an integer, c sets of k rows by v columns are obtained. Each set is an optimal row-column design. To obtain a printed copy of the output for v = 9, k = 3, and r = 4, use the command

CIB v=9 k=3 r=4 TRIES=100
COPY CIB.OUT PRN (or PRINT CIB.OUT)

The printed output obtained is

```
NOTE: Construct a CIBD with v=9, k=3, r=4 and b=12

 rand_seed        E     E/U  11    n1    n2    n3    n4
 451269391   698361   93.11   0     2     4     2
 358268564   704471   93.93   0     2     4     2

NOTE: Initial block(s) obtained at try 2:

   8   1   4
   1   4   7

NOTE: Initial block(s) were used to construct the
following design:

   3   1   4   3   8   5   7   6   1
   9   3   6   8   4   2   4   2   5
   7   6   9   5   1   9   2   8   7

   1   6   8
   7   3   2
   4   9   5

NOTE: Blocks are columns (block order randomized).
NOTE: Treatments within blocks are randomized.
NOTE: CIB used -0.00 seconds.
```

When k is an integer multiple of r, i.e., 3, 6, 9,..., sets of k = 3 rows by v=9 columns will appear. Each set is an optimal row-column design of the Youden design type. Here a three concurrence design was obtained with E = 0.704471 rather than the E = 0.75 obtained from BIB. The concurrences of one of the treatments is given, and n1 + n2 + n3 + n4 = v - 1.

For the second example, consider the following commands and output:

CIB v=12 k=3 r=4 TRIES=100
COPY CIB.OUT PRN (or PRINT CIB.OUT)

```
NOTE: Construct a CIBD with v=12, k=3, r=4 and b=16

 rand_seed        E    E/U  l1    n1    n2    n3    n4
 817003809   703164  99.08   0     3     8

NOTE: Initial block(s) obtained at try 1:

   2   5   7
   4   8  12

NOTE: Initial block(s) were used to construct the
following design:

   3   4  10   9   7   1   6   7   1   4   9   6
  12   2   8   2   2  11   1  10   3   7   6   8
   5  11   5  12   5   8   4  12  10   9  11   3

  10   4   9   7
   2  12   5  11
   6   8   1   3

NOTE: Blocks are columns (block order randomized).
NOTE: Treatments within blocks are randomized.
NOTE: CIB used 0.05 seconds.
```

This is a two concurrence design with $\lambda_1 = 0$ and $\lambda_2 = 1$. A given treatment appears with 8 treatments once and does not appear with 3 other treatments. E/U for this design is 99.08, not as efficient as the design obtained with BIB. n1 + n2 + n3 + n4 = v – 1 = 8 + 3 =11.

6. ROW-COLUMN MODULE

An optimal or near optimal incomplete block design is obtained as the first step in obtaining the row-column design. The resulting design is entered and then the optimal or near optimal row-column design is obtained. The first example is a k = 4 row by s = 7 column design for v = 7. The resulting design will be a Youden design and is obtained with the following commands

BIB v=7 k=4 r=4 –c Tries=20
COPY BIB.OUT=CAR.IN
EDIT CAR.IN

Or an alternative is

BIB v=7 k=4 r=4 –c TRIES=20
EDIT BIB.OUT

The editing is done by removing all text and all blank lines, leaving only the IBD. Then from the FILE menu, select SAVE to save the edited CAR.IN (or edited BIB.OUT). Select EXIT from the FILE menu.

C:\Gendex>CAR IN=CAR.IN (or BIB.OUT)
C:\Gendex>COPY CAR.OUT PRN

The printed output is

```
NOTE: Construct an RCD with v=7, k=4, b=7 and r=4

 rand_seed          f  E(ROW)  E(COL)  E(RCD)     E/U
 953184785      18900  1.0000  0.8750  0.8750  100.00

   4   2   5   7   3   6   1
   3   6   2   1   4   7   5
   6   5   7   3   1   2   4
   5   1   3   6   2   4   7

NOTE: The above design was obtained at try 3.
```

NOTE: CAR used 0.00 seconds.

For a second example, let v=21treatments are arranged in a five row by 21 column design. To produce these designs, use the following commands

BIB v=21 k=5 r=4 –c TRIES=20
EDIT BIB.OUT

The editing is done by removing all text and all blank lines, leaving only the IBD. Then from the FILE menu, select SAVE to save the edited BIB.OUT. Select EXIT from the FILE menu.

CAR IN=BIB.OUT
COPY CAR.OUT PRN

The printed output is

```
NOTE: Construct an RCD with v=21, k=5, b=21 and r=5

 rand_seed          f  E(ROW)  E(COL)  E(RCD)     E/U
 148127391     444360  1.0000  0.8400  0.8400  100.00

   7  13  11   2  14  17  18   3   9  15  10   4   1  21   8   6  12  20  16  19   5
  18   1  19  21  15   6  14   9   2   5   3   8  16  20  10  13   4   7  11  12  17
  16   7  15  19  13   9   1   8  12  10  21   2  20  18  14   4   5   3   6  17  11
  17  21   9   5  16  14   2  18  10   1  11  15   4  12  20  19   7   6   8   3  13
  10   8   7  16   3  21  11   5  13   6   4  17   9  15  19  18  14   2  12   1  20

NOTE: The above design was obtained at try 37.
NOTE: CAR used 0.61 seconds.
```

This design is a Youden design and, as a result, E/U = 100.00.

7. RESOLVABLE ROW-COLUMN DESIGN MODULE

To obtain a resolvable row-column design, one needs to first construct and IBD using BIB or ALPHA. To illustrate, use the command

BIB v=9 k=3 r=4 –r –c TRIES=10

This command creates a resolvable IBD with columns (-c) as the incomplete blocks. Then use the command

COPY BIB.OUT=RRC.IN

RRC.IN needs to be edited before it can be used. To do this, use the command

EDIT RRC.IN

The editing is done by removing all text and all blank lines, leaving only the IBD. Then from the FILE menu, select SAVE to save the edited RRC.IN. Select EXIT from the FILE menu. To obtain a resolvable row-column design for nine treatments in three rows and three columns in each complete block (a balanced lattice square design is obtained), use the command

RRC in=RRC.IN

This command produces the resolvable row-column design. To display the design on the monitor, use the command

TYPE RRC.OUT

To obtain a printed copy, use one of the commands

PRINT RRC.OUT or COPY RRC.OUT PRN

The latter command may copy the output somewhere in the computer. Then, after finding it, a printed copy may be obtained. The printed output from the above command is

```
NOTE: Construct an RCD with v=9, r=4, k=3 and s=3

 rand_seed          f  E(ROW)  E(COL)  E(RCD)     E/U
 998572763       1296  0.7500  0.7500  0.5000  100.00

   9   1   3
   7   4   6
   5   2   8

   6   8   3
   5   7   1
   2   4   9

   8   9   7
   5   4   3
   1   6   2

   9   5   6
   2   3   7
   8   4   1

NOTE: The above design is obtained at try 2.
NOTE: RRC used 0.00 seconds.
```

For the second illustration, we use v=12 treatments arranged in three rows and four columns in each of four complete blocks. The commands are

BIB v=12 k=3 r=4 -r -c TRIES=100
EDIT BIB.OUT (edit as before, SAVE, and EXIT from FILE menu)
RRC.IN=BIB.OUT
COPY RRC.OUT PRN

The resulting printed copy is

```
NOTE: Construct an RCD with v=12, r=4, k=3 and s=4

 rand_seed          f  E(ROW)  E(COL)  E(RCD)     E/U
1162777652       2604  0.7891  0.7097  0.5346   99.53

   6   2   7   9
  12  11   3   1
  10   4   5   8

   4   9   5  11
   1   2   6  10
  12   3   8   7

   4  10  12   9
   8   3   2  11
   7   1   5   6

   6  11   8  12
   4   1   2   7
   3   5  10   9

NOTE: The above design is obtained at try 76.
NOTE: RRC used 0.38 seconds.
```

E/U was 99.53, which is nearly at the maximum level of 100.00. As noted, this design was obtained at try 76. Whether or not a design with E/U = 100.00 could have been obtained was not determined even though the process was allowed to run for ten minutes.

8. REMARKS

As noted above, different designs may be obtained with the different modules. For any incomplete block design which does not have E/U = 100.00 more tries or another module should be tried and the design resulting in the highest value of E/U should be selected. The same procedure should be followed when constructing row-column or resolvable row-column designs.

For the CIB subroutine, one may obtain the design with treatments NOT randomized within the blocks by using the command -y. However, the order of the blocks will be randomized. The command -y may be placed anywhere in the command statement in the same manner as for -c and -r.

The output may be renamed if another name is desired. For example, a row-column design could be renamed from RRC.OUT to ROWCOL.OUT using the command:

RRC OUT=ROWCOL.OUT

It is a good idea to put in the number of tries in order to keep the program from running for long periods of time. A program may be stopped by pressing any key if the number of tries is not stated and if the program is still attempting to find a design with E/U = 100.00. Also, it is to be noted that two different starts for the same design may give two different designs with different E/U values for a specific number of tries. When E/U is not high enough for one run, another run or another module for the same values of v, k, and r should be conducted.

Another feature of Gendex is that an existing design may be augmented with additional replicates. Suppose that v = 15, k = 3, and r = 2 is the initial design but it is desired to add an additional replicate. The program for doing this is

```
BIB v=15 k=3 r=2 TRIES=30 -r
COPY BIB.OUT=BIB.IN
EDIT BIB.IN (Delete all text and blank lines, SAVE and EXIT)
BIB v=15 k=3 r0=2 r=3 TRIES=30 -r TRIES=30
COPY BIB.OUT PRN (or PRINT BIB.OUT)
```

The r0 in the next to last command is the number of replicates in the original design.

If one types GENDEX while in the Gendex directory, the following appears

Module name OVERWRITE [Y/N]

Simply type N for each of the questions as the list of files appears on the monitor. With this command one is able to view which files appear in the toolkit.

9. SOME RELEVANT LITERATURE

Miller, A. J. and N-K Nguyen (1994). A Federov exchange algorithm for D-optimal design. Applied Statistics 43(4), 669-678.

Nguyen, N-K (1992). A review of some exchange algorithms for constructing discrete D-optimal designs. Computational Statistics & Data Analysis 14, 489-498.

Nguyen, N-K (1993). An algorithm for constructing optimal resolvable incomplete block designs. Communications in Statistics – Simulation 22(3), 911-923.

Nguyen, N-K (1994). Construction of optimal block designs by computer. Technometrics 36(3), 300-307.

Nguyen, N-K (1996). A note on the construction of near-orthogonal arrays with mixed levels and economic run size. Technometrics 38:279-283.

Nguyen, N-K (1996). An algorithmic approach to constructing supersaturated designs. Technometrics 38(1), 69-73.

Nguyen, N-K (1997). Construction of optimal row-column designs by computer. Computing Science & Statistics 28:471-474.

Nguyen, N-K (1998). Cutting experimental designs into blocks. CMIS Technical Report 98/80.

Nguyen, N-K and A. J. Miller (1997). 2^m fractional factorial designs of resolution V with high A-efficiency, $7 \leq m \leq 10$. J. Statistical Planning and Inference 59, 379-384.

Nguyen, N-K and E. R. Williams (1993). An algorithm for constructing optimal resolvable row-column designs. Australian Journal of Statistics 35(3), 363-379.